

--	--	--	--	--	--	--	--

Vpisna številka

1	
2	
3	
Σ	

NAVODILA

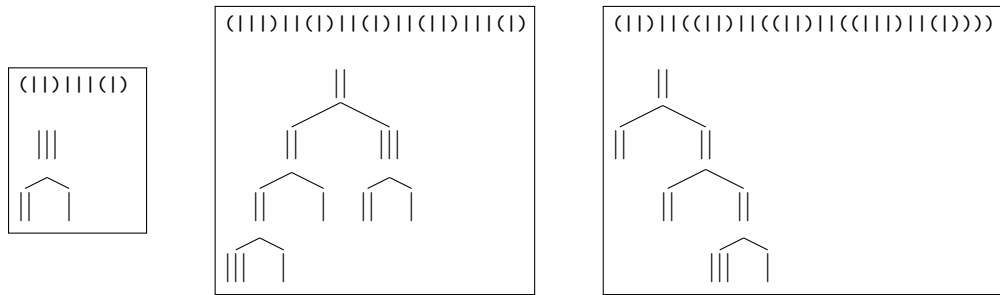
- **Ne odpirajte te pole**, dokler ne dobite dovoljenja.
- **Preden začnete reševati test:**
 - Vpišite svoje podatke na testno polo z velikimi tiskanimi črkami.
 - Na vidno mesto položite osebni dokument s sliko in študentsko izkaznico.
 - Preverite, da imate mobilni telefon izklopljen in spravljen v torbi.
 - Prijavite se na spletno učilnico, kamor boste oddajali nekatere odgovore.
- Dovoljeni pripomočki: pisalo, brisalo, USB ključ, in poljubno pisno gradivo.
- Rešitve vpisujte v polo ali jih oddajte preko spletne učilnice. Pri odgovorih, ki ste jih oddali preko spletne učilnice, na izpitno nalogo napišite "glej spletno učilnico – datoteka <ime_datoteke>".
- Če kaj potrebujete, prosite asistenta, ne sosedov.
- **Med izpitom ne zapuščajte svojega mesta** brez dovoljenja.
- Testna pola vam bo odvzeta **brez nadaljnjih opozoril**, če:
 - komunicirate s komerkoli, razen z asistentom,
 - komu podate kak predmet ali list papirja,
 - odrinete svoje gradivo, da ga lahko vidi kdo drug,
 - na kak drug način prepisujete ali pomagате komu prepisovati,
 - imate na vidnem mestu mobilni telefon ali druge elektronske naprave.
- **Ob koncu izpita:**
 - Ko asistent razglasi konec izpita, **takoj** nehajte in zaprite testno polo.
 - **Ne vstajajte**, ampak počakajte, da asistent pobere vse testne pole.
 - **Testno polo morate nujno oddati.**
- Čas pisanja je 120 minut. Na vidnem mestu je zapisano, do kdaj imate čas.
- Predvideni ocenjevalni kriterij:
 1. ≥ 90 točk, ocena 10
 2. ≥ 80 točk, ocena 9
 3. ≥ 70 točk, ocena 8
 4. ≥ 60 točk, ocena 7
 5. ≥ 50 točk, ocena 6

Veliko uspeha!

1. naloga (35 točk)

a) (7 točk)

V Elboniji so arheologi odkrili 5000 let stare glinene tablice:



V bližini so odkrili še glineno tablico, na kateri je v praelbonščini pisalo:

Vsaka od priloženih glinenih tablic ima na vrhu vklesan izraz in pod njim pripadajoče sintaktično drevo. Na tablico za odgovore vklešite slovnico, ki dane izraze pretvori v pripadajoča sintaktična drevesa. (Veljavnih rešitev je več.)

Nalogo rešite tudi vi. Pravila zapišite v obliki BNF.

b) (14 točk) V Ocamlu implementirajte modul `Trilean`, ki ustreza naslednji signaturi `TRILEAN`:

```
module type TRILEAN =
sig
  type tri = True | False | Probably of float
  type hidden
  val conjunction : tri * tri -> tri
  val negation : tri -> tri
  val congregation : (tri * tri -> tri) -> (tri * tri -> tri) -> (tri * tri -> tri * tri)
  val hide : tri -> hidden
  val reveal : hidden -> tri
end
```

Kaj vaša implementacija počne, ni pomembno, mora pa zadoščati dani signaturi.

c) (7 točk) V funkcijskem programskem jeziku s parametričnim polimorfizmom izračunajte glavni tip funkcije f :

```
let rec f g = function
  | (h1::t1, h2::t2) -> g h1 h2 ~ f g (t1, t2)
  | _ -> ""
```

Opomba: v OCamlu je \sim operator za stikanje nizov znakov. Za vse točke je potreben postopek.

d) (7 točk) V λ -računu definiramo izraze:

$$\begin{aligned} I &:= \lambda y. y \\ S &:= \lambda x. \lambda y. \lambda z. x z (y z) \\ K &:= \lambda x y. x \end{aligned}$$

Izračunajte normalno obliko (vrednost) izraza $I S K K (K I)$. Za vse točke je potreben postopek.

2. naloga (40 točk)

a) (20 točk) Ukazni programski jezik razširimo s funkcijo $f : \mathbb{Z} \rightarrow \mathbb{Z}$. Dokažite *delno* pravilnost programa:

```
{ n ≥ 1 }  
m := f(1) ;  
i := 2 ;  
while i ≤ n do  
  if m < f(i) then  
    m := f(i) ;  
    i := i + 1  
  else  
    skip  
  end  
done  
{ m = max(f(1), f(2), ..., f(n)) }
```

Pišite čitljivo!

b) (5 točk) Naj bo P zgornji program. Ali je zadoščeno popolni pravilnosti

$$[n \geq 1] P [m = \max(f(1), f(2), \dots, f(n))] ?$$

Odgovor utemeljite.

c) (15 točk) Program P implementirajte kot funkcijo v OCamlu ali Haskellu

```
najvecji : (int -> int) -> int -> int
```

kjer `najvecji` f n vrne $\max(f(1), \dots, f(n))$, če velja $n \geq 1$. Primer:

```
# najvecji (fun k -> k * (20 - k)) 18 ;;
- : int = 100
# najvecji (fun k -> 2 * k + 8) 100 ;;
- : int = 208
```

Za vse točke uporabite repno rekurzijo.

3. naloga (35 točk)

Pravimo, da je matrika

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix}$$

naraščajoča, če vrednosti naraščajo, ko se premikamo desno in navzdol, se pravi, da za vse $1 \leq i < m$ in $1 \leq j < n$ velja $a_{i,j} < a_{i+1,j}$ in $a_{i,j} < a_{i,j+1}$.

Naloge rešujte v prologu in uporabite programiranje z omejitvami v domeni `clpfd`.

a) (10 točk)

Sestavite predikat `narascajocaVrstica(V)`, ki ugotovi, ali je `V` naraščajoč seznam. Primeri:

```
?- narascajocaVrstica([1,5,6,10]).
true.
?- narascajocaVrstica([1,5,5,10]).
false.
?- narascajocaVrstica([1,A,B,10]).
A in 2..8,
A#=<B+ -1,
B in 3..9.
```

b) (25 točk)

Sestavite predikat `narascajoca(M)`, ki ugotovi, ali je dana matrika `M` naraščajoča. Matriko predstavimo s seznamom vrstic in predpostavimo, da vsebuje cela števila. Seveda lahko sestavite še ustrezne pomožne predikate. Primeri:

```
?- narascajoca([[1,2,5], [2,4,6]]).
true.
?- narascajoca([[1,2,5], [2,4,3]]).
false.
?- narascajoca([[1,A,B,4], [3,C,6,D]]).
A = 2,
B = 3,
C in 4..5,
D in 7..sup.
?- narascajoca([[1,A,B,4], [3,C,8,D], [10,20,F,G]]).
A = 2,
B = 3,
C in 4..7,
D in 9..sup,
D#=<G+ -1,
F in 21..sup,
F#=<G+ -1,
G in 22..sup.
```